Discovering Long-period Exoplanets using Deep Learning with Citizen Science Labels

Shreshth A. Malik[*]
OATML
University of Oxford

Nora L. Eisner
Department of Physics
University of Oxford

Chris J. Lintott
Department of Physics
University of Oxford

Yarin Gal
OATML
University of Oxford

Abstract

Automated planetary transit detection has become vital to prioritize candidates for expert analysis given the scale of modern telescopic surveys. While current methods for short-period exoplanet detection work effectively due to periodicity in the light curves, there lacks a robust approach for detecting single-transit events. However, volunteer-labelled transits recently collected by the Planet Hunters TESS (PHT) project now provide an unprecedented opportunity to investigate a data-driven approach to long-period exoplanet detection. In this work, we train a 1-D convolutional neural network to classify planetary transits using PHT volunteer scores as training data. We find using volunteer scores significantly improves performance over synthetic data, and enables the recovery of known planets at a precision and rate matching that of the volunteers. Importantly, the model also recovers transits found by volunteers but missed by current automated methods.

1 Introduction

Astronomical datasets from recent survey missions such as TESS [1] are now large enough to make inspection by experts implausible. Discovering planets by observing the effect of their passage on their parent stars—the transit method—for example, requires automated analysis. However, incumbent algorithms that flag potential exoplanets rely on detecting periodic signals in the brightness of the star—its light curve (LC). This skews the observed distribution to short-period planets, resulting in fewer longer-period planets than expected in our catalog. In response, citizen science projects have proven successful in finding planets that have been missed by automated detection methods [2, 3]. In Planet Hunters TESS[2] (PHT), volunteers from the general public inspect LCs for planetary transits by eye. The data flagged by the volunteers has enabled a number of novel astronomical discoveries, notably of longer-period exoplanets [3, 4].

In this work, we investigate using volunteer transit confidence scores from PHT as soft labels to train a 1-D convolutional neural network (CNN) to detect single-transit events from TESS LCs. We find that training with volunteer scores as the main training signal enables better recovery of known planets compared to synthetic data, with a precision and recall similar to that of the volunteers. Moreover, the model is able to detect planets missed by traditional automated algorithms and even some that are missed by volunteers. The model could therefore serve as a basis for a human-in-the-loop machine learning pipeline for longer-period exoplanet discovery.

[*]Correspondance to shreshth@robots.ox.ac.uk

[2]http://www.planethunters.org

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

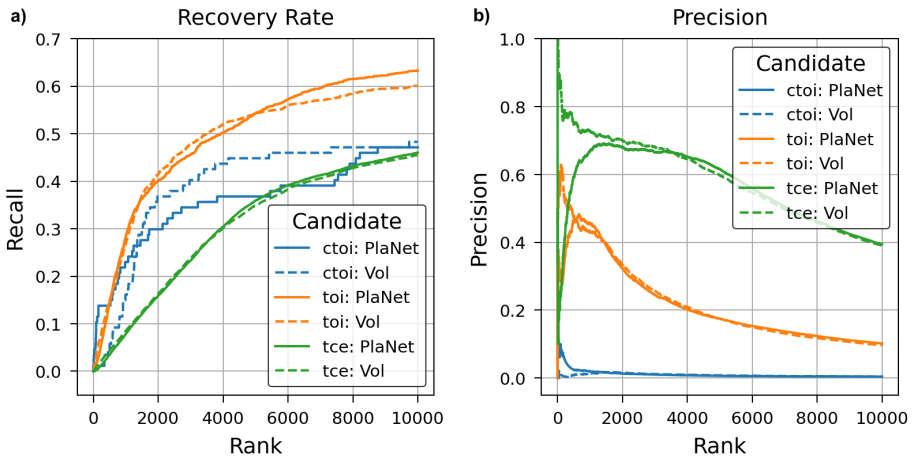

Figure 1: Known candidate recovery for the PlaNet model and volunteers on test sectors. **a)** Fractional recovery rate when test samples are ranked in order of predicted confidence. **b)** Precision of the top k ranked predictions.

2 Background and Related Work

We focus on the *detection* part of the exoplanet discovery pipeline, which involves identifying and short-listing high-potential exoplanet candidates from a large observational dataset for further analysis. We investigate detecting exoplanets using the transit method, where temporary decreases in brightness of stars can be used to identify a transiting planet.

Light Curves The observed brightness of a star varies due to both astrophysical and instrumental noise processes. These variations can include both periodic (e.g. stellar variability) and irregular signals (e.g. flares) which can either be confused for transits, or can confound automatic searches. Contamination from other sources, the inherent variability of the target stars and telescope systematics can also cause anomalies and missing data. This means that classical signal processing struggles to robustly identify transits across the wide distribution of observed LCs. Figure 3 in Appendix A shows some examples of typical LCs.

Traditional Exoplanet Detection and Validation The most common approach to automated detection is to search for periodic signals and use a box-fitting algorithm to find transit-like dips in the phase-folded LC [5]. The flagged LCs are then validated using a combination of diagnostics, human vetting, probabilistic and machine learning approaches [6, 7, 8, 9, 10, 11, 12, 13, 14].

Deep Learning for Exoplanet Detection Recent work has sought to go beyond validation and instead use deep learning directly on the (non-phase-folded) LCs to discover new candidates missed by detection algorithms. However, the comparatively small number of positive examples has limited the applicability of deep learning for exoplanet discovery [15]. Prior work trains on simulated data to overcome this bottleneck, but this has been shown to have limited applicability on real LCs due to the complex noise processes involved [8, 16]. Olmschenk et al. [17] used a combination of real and synthetic data to construct a pipeline from full-frame images rather than the LCs. Nonetheless, the authors note that they still find a similar distribution of planets to those found by automated algorithms because the training data is biased towards multi-transit events. Cui et al. [18] present a promising approach of using 2-D object detection algorithms on images of the LC, but are again limited in training data by transits found by automated algorithms. In our work, by leveraging volunteer scores, we are able to recover planets outside of the distribution found by automated algorithms.

Table 1: Recovery of TOIs comparing volunteers to variations of PlaNet. The subscript denotes the proportion of synthetic planets and EBs used (default, 0.1). The recall (R) and precision (P) for the top k ranked predictions, and the receiver operating characteristic and precision-recall area under curve (AUC, PR-AUC) metrics are given for each model.

Model	**TOIs**									
	R@100	1,000	5,000	10,000	P@100	1,000	5,000	10,000	AUC	PR-AUC
Volunteers	**0.030**	0.268	**0.543**	0.601	**0.490**	0.431	**0.175**	0.097	0.799	**0.261**
PlaNet	0.019	**0.283**	**0.542**	**0.633**	0.300	**0.455**	**0.175**	**0.102**	**0.835**	0.227
$\text{PlaNet}_{0.0}$	0.023	0.241	0.525	0.594	0.370	0.388	0.169	0.096	0.810	0.197
$\text{PlaNet}_{0.3}$	0.009	0.158	0.518	0.620	0.140	0.255	0.167	0.100	0.637	0.147
$\text{PlaNet}_{\text{all}}$	0.009	0.113	0.396	0.517	0.140	0.182	0.127	0.083	0.756	0.091

3 Data

3.1 TESS Light Curves

The TESS mission observes around 20,000 stars from a new sector of the sky every month. We used the two-minute cadence LCs from the SPOC pipeline [19][3]. To provide more realistic evaluation, we divided the train and test data by sector as the model will be used for prioritizing future observations rather than post-hoc analysis. Sectors 10-35 were used for training and validation, and sectors 36-43 for evaluation. Appendix B.1 contains more details on the data.

Pre-processing The LCs were pre-processed in three steps. **1) Anomaly removal**. We used the PDCSAP fluxes, which are nominally corrected for instrument variations and flux contamination from nearby stars. We also filtered using the `QUALITY` marker given in LC files to remove anomalies. **2) Binning**. For computational reasons we binned the data to a 14-minute cadence. This is the same binning factor for LCs shown to the volunteers. The duration of transits of interest are generally on the order of hours to days, so the characteristic shape of the dip is still identifiable at this resolution. Empirical tests confirmed that there was no significant performance difference in using LCs binned to 6-minutes or 14-minutes. **3) Normalisation**. We divided and subtracted by the median such that the LC is centred at zero. Dividing by the median (rather than the standard deviation) is common practice in LC analysis as this allows a comparison of the magnitude of brightness dips, which can differentiate between false positives and planets. We truncated from the start and end (in random proportions) such that all LCs have a binned length of 2,500, and imputed missing values with zeros.

Synthetic Data In this work we investigate the efficacy of using volunteer scores as a training signal for single-transit detection. We compare model performance when trained with varying amounts of additional synthetic data as a comparison. The majority of transit-like signals in LCs correspond to false positives ($> 95\%$) [20]. Thus to help differentiate these cases, we included a proportion of synthetic data from the test ETE-6 dataset [21] corresponding to transits and eclipsing binaries (EBs, the most common false positive) in equal proportions. As we focus on single-transits, we only take one transit from each synthetic LC to inject into a random section of the base LC. We used the full flux for EBs as asymmetric dips are often used to identify them.

3.2 Volunteer Scores and Planetary Candidate Labels

We used the aggregated confidence scores from PHT that take into account the volunteer skill level [3] as soft targets to train the network. We also cross-referenced the star corresponding to each LC with discrete classifications of planetary candidates from TESS for evaluation. The TESS automated pipeline flags $\sim 7\%$ of the observations as *threshold crossing events* (TCEs) which indicates the presence of a periodic signal. TCEs are then analysed by the TESS team and $\sim 18\%$ are promoted to *TESS Objects of Interest* (TOIs) status as likely planetary candidates. Similarly, candidates that have been identified through volunteer flagging and vetted by the PHT science team are called *Community*

[3] `https://archive.stsci.edu/missions-and-data/tess`

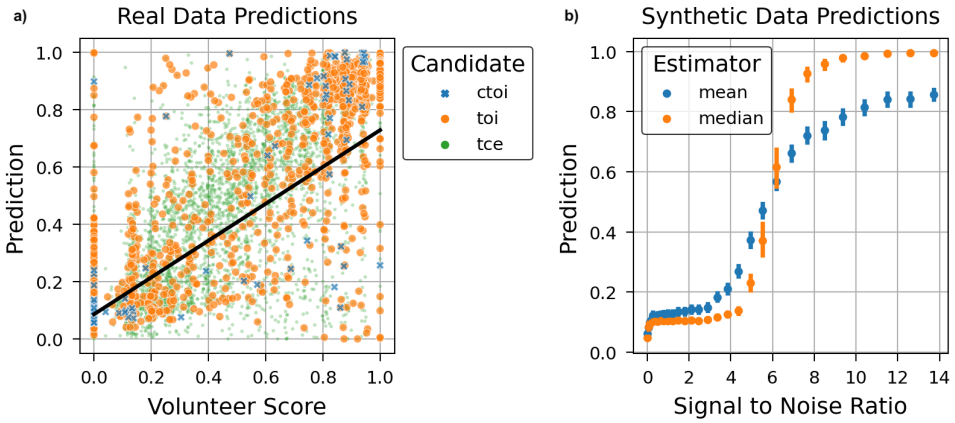

Figure 2: Predictions on real and synthetic test data. **a)** Parity plot for PlaNet predictions on real data. Null label predictions from volunteer scores are not plotted for clarity but included in the regression computation (black line). **b)** Predictions on injected transits as a function of signal to noise ratio. A 95% confidence interval is shown for each binned value.

TOIs (cTOIs). Overall, $\sim$1% of all LCs have planetary transits, whereas $\sim$21% have a non-zero volunteer confidence scores. Thus the soft labels provide a better training signal to the model.

4 Method

PlaNet Model A 1-D CNN we call *PlaNet* was trained using a cross-entropy loss for binary classification of light curves with or without a planetary transit. The model is primarily based on the work by Olmschenk et al. [17]. In comparison to Olmschenk et al. [17], we used a deeper network (22 convolutional layers) and larger kernel sizes in the first two layers (7 and 5 respectively) without down-sampling. We also used residual connections to mitigate the vanishing gradient effect that is associated with deeper models [22]. These modifications resulted in significant performance gains over the original model. Appendix B.2 contains further details on the model. Our code is publicly available[4].

Data Augmentations During training, we added transformations to help with generalisation, each with a probability of 0.1. To simulate a noise process, we randomly chose another LC with volunteer score of 0.0 to inject into the base LC. Three types of data shifts were also incorporated. 1) Two randomly chosen non-overlapping sections (each 25% of the LC) were switched. 2) The LC was reversed temporally. 3) A random section (10%) of the LC was deleted.

5 Results and Discussion

Using volunteer scores as training data enables effective and diverse planet recovery. Figure 1 shows the fractional recovery rate and precision of planetary candidates for the top $\sim 10\%$ ranked predictions of PlaNet and volunteers. We find that PlaNet's performance on TOIs is similar to that of volunteers. Moreover, we find that the model recovers around a third of the cTOIs in the top 2% ranked results, indicating that the model also finds planets that the TOI pipeline misses. Evaluation on synthetic data (Figure 2b) indicates that performance greatly increases for transits with a signal to noise ratio greater than 6. This is similar to that observed for volunteers [3].

Synthetic data can be useful but is not enough. Table 1 shows TOI recovery performance of PlaNet when trained with different proportions of synthetic data. We find that training with volunteer scores significantly increases performance over training with only synthetic data. However the best performing model is one that does leverage a small amount of synthetic data (10% synthetic transits, 10% synthetic EBs, 80% volunteer soft labels). We argue that the volunteer labels provides a strong

[4] `https://github.com/s-a-malik/pht-ml`

training signal, but the additional hard labels provided by the synthetic data help identify transits and rule out false positives with more confidence. This leads to improved TOI recovery.

Comparing PlaNet and volunteer predictions. In general, we find that the TOIs and cTOIs have high model predictions (Figure 2a)[5]. In some cases however, there is disagreement between volunteer and model predictions. A review of a selection of these disputed LCs by an expert author (NE) suggested that several ($\sim 30\%$) of those identified by PlaNet but missed by volunteers were worth further investigation, suggesting that the network is providing suitable candidates for review and observational follow-up. We also found that fewer multi-transit planets were recovered, likely as we explicitly focus training on single-transit events. Appendix A contains further qualitative analysis.

6 Conclusion

In this work, we used volunteer scores from Planet Hunters TESS to train a 1-D CNN to detect planetary transits from TESS light curves. The model was found to match the original volunteer's recovery rate. Moreover, as observed in Eisner et al. [3], we recover single-transit candidates that are missed by traditional pipelines, and even some that are missed by volunteers. The model could therefore serve as a basis for a human-in-the-loop pipeline for longer-period exoplanet discovery. In future work, we hope to use the model predictions alongside volunteer scores on new sectors to prioritize analysis. Incorporating calibrated uncertainty estimation would help identify costly overconfident predictions. This will also enable an active learning pipeline to be developed which can make better use of volunteer time [23, 24].

Broader Impact

Citizen science enables more accessible participation in the scientific process. We believe involving the public in scientific endeavours is beneficial for the community, widens perspectives and accelerates progress. We see our work as a step towards human-in-the-loop machine learning pipeline where volunteer and expert time for tedious labelling and analysis tasks is reduced. However, we must also note that systems trained on crowd-sourced data can amplify existing biases if they exist in the data. If there are systematic issues with volunteer labelling these can propagate to the model and potentially be detrimental to performance.

Acknowledgments and Disclosure of Funding

SM acknowledges funding from EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems (Grant No: EP/S024050/1). YG acknowledges funding from the Turing Fellowship (Grant No. EP/V030302/1). Some of the data presented in this paper was obtained from the Mikulski Archive for Space Telescopes (MAST). Planet Hunters TESS is supported in part by the Alfred P. Sloan Foundation.

References

[1] George R Ricker et al. "Transiting exoplanet survey satellite". In: *Journal of Astronomical Telescopes, Instruments, and Systems* 1.1 (2014), p. 014003.

[2] Debra A Fischer et al. "Planet Hunters: the first two planet candidates identified by the public using the Kepler public archive data". In: *Monthly Notices of the Royal Astronomical Society* 419.4 (2012), pp. 2900–2911.

[3] Nora L Eisner et al. "Planet Hunters TESS II: findings from the first two years of TESS". In: *Monthly Notices of the Royal Astronomical Society* 501.4 (2021), pp. 4669–4690.

[4] Nora L Eisner et al. "Planet Hunters TESS IV: a massive, compact hierarchical triple star system TIC 470710327". In: *Monthly Notices of the Royal Astronomical Society* 511.4 (2022), pp. 4710–4723.

[5] Note that the majority of (c)TOIs with both low volunteer and low model scores do not have transits in that particular LC. Instead these are identified when the same star is observed in other sectors. The results in Table 1 are thus an underestimate of true performance.

[5] Geza Kovács, Shay Zucker, and Tsevi Mazeh. "A box-fitting algorithm in the search for periodic transits". In: *Astronomy & Astrophysics* 391.1 (2002), pp. 369–377.

[6] Benjamin T Montet et al. "Stellar and planetary properties of K2 campaign 1 candidates and validation of 17 planets, including a planet receiving earth-like insolation". In: *The Astrophysical Journal* 809.1 (2015), p. 25.

[7] Susan E Thompson et al. "A machine learning technique to identify transit shaped signals". In: *The Astrophysical Journal* 812.1 (2015), p. 46.

[8] Shay Zucker and Raja Giryes. "Shallow transits—deep learning. I. Feasibility study of deep learning to detect periodic transits of exoplanets". In: *The Astronomical Journal* 155.4 (2018), p. 147.

[9] Christopher J Shallue and Andrew Vanderburg. "Identifying exoplanets with deep learning: A five-planet resonant chain around kepler-80 and an eighth planet around kepler-90". In: *The Astronomical Journal* 155.2 (2018), p. 94.

[10] Megan Ansdell et al. "Scientific domain knowledge improves exoplanet transit classification with deep learning". In: *The Astrophysical journal letters* 869.1 (2018), p. L7.

[11] Liang Yu et al. "Identifying exoplanets with deep learning. III. Automated triage and vetting of TESS candidates". In: *The Astronomical Journal* 158.1 (2019), p. 25.

[12] Hugh P Osborn et al. "Rapid classification of TESS planet candidates with convolutional neural networks". In: *Astronomy & Astrophysics* 633 (2020), A53.

[13] David J Armstrong, Jevgenij Gamper, and Theodoros Damoulas. "Exoplanet validation with machine learning: 50 new validated Kepler planets". In: *Monthly Notices of the Royal Astronomical Society* 504.4 (2021), pp. 5327–5344.

[14] Hamed Valizadegan et al. "ExoMiner: A Highly Accurate and Explainable Deep Learning Classifier That Validates 301 New Exoplanets". In: *The Astrophysical Journal* 926.2 (2022), p. 120.

[15] Trisha A Hinners, Kevin Tat, and Rachel Thorp. "Machine learning techniques for stellar light curve classification". In: *The Astronomical Journal* 156.1 (2018), p. 7.

[16] Abhishek Malik, Benjamin P Moster, and Christian Obermeier. "Exoplanet detection using machine learning". In: *Monthly Notices of the Royal Astronomical Society* 513.4 (2022), pp. 5505–5516.

[17] Greg Olmschenk et al. "Identifying Planetary Transit Candidates in TESS Full-frame Image Light Curves via Convolutional Neural Networks". In: *The Astronomical Journal* 161.6 (2021), p. 273.

[18] Kaiming Cui et al. "Identify Light-curve Signals with Deep Learning Based Object Detection Algorithm. I. Transit Detection". In: *The Astronomical Journal* 163.1 (2021), p. 23.

[19] Jon M Jenkins et al. "The TESS science processing operations center". In: *Software and Cyberinfrastructure for Astronomy IV*. Vol. 9913. SPIE. 2016, pp. 1232–1251.

[20] Peter W Sullivan et al. "The Transiting Exoplanet Survey Satellite: simulations of planet detections and astrophysical false positives". In: *The Astrophysical Journal* 809.1 (2015), p. 77.

[21] Jon M Jenkins et al. "A Simulated Data Set for the Transiting Exoplanet Survey Satellite". In: *Research Notes of the AAS* 2.1 (2018), p. 47.

[22] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[23] Burr Settles. "Active learning literature survey". In: (2009).

[24] Mike Walmsley et al. "Galaxy Zoo: probabilistic morphology through Bayesian CNNs and active learning". In: *Monthly Notices of the Royal Astronomical Society* 491.2 (2020), pp. 1554–1574.

[25] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[26] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.

[27] Adam Paszke et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019).

[28] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5 and Appendix A.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Broader Impact section
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See `https://github.com/s-a-malik/pht-ml`
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 and Appendix B
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Uncertainty quantification will be done in further work (Section 6)
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix B.2
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 3 and Section 4
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] Data was crowdsourced by the Planet Hunters TESS project. See Eisner et al. [3] for details.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Qualitative Analysis

We can gain an insight into where PlaNet does well and where it fails by looking at example predictions on the test sectors. Concretely, we investigated cases where there are the largest discrepancies between model predictions and volunteer scores (i.e. the top left and bottom right corners of Figure 2a). Figure 3 shows examples of these cases.

PlaNet Success Modes Figure 3a shows a case where volunteers rejected a promising candidate (likely due to the low signal to noise ratio), but PlaNet successfully flagged it as a likely transit. Figure 3b shows a false positive from the volunteers. Here the dip is due to a background event, but may be mistaken for a transit to the untrained eye. This was correctly predicted to be an unlikely transit by the model.

PlaNet Failure Modes The bottom row of Figure 3 shows examples where volunteers correctly identify planetary candidates and false positives, whereas the model makes incorrect predictions with high confidence. We find the model sometimes rejects candidates with multi-transit events. As we have explicitly trained on single-transit synthetics, the model may mistake slight asymmetry in multi-transit dips as an eclipsing binary (Figure 3c). We also find that false positives arise more often when there is a lot of stellar variability, as the model may mistake the sharp intrinsic variation for a transit (Figure 3d).

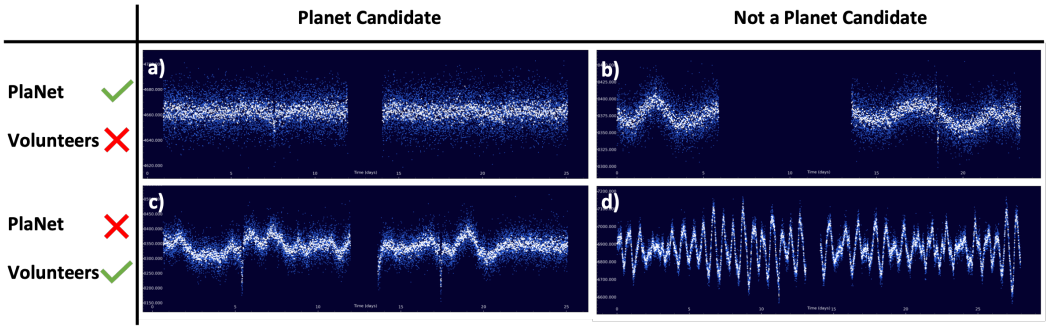

Figure 3: Example success and failure modes of volunteers and the PlaNet model. In particular, we look at examples of both false positive and false negative prediction of known planets, where PlaNet and the volunteers' predictions are the most different. Each panel shows binned (white) and un-binned (blue) TESS light curve data. **a)** An example of a PlaNet true positive and volunteer false negative. Here PlaNet flagged a new potential candidate with a shallow dip that volunteers missed. *TIC ID: 371596256, Sector: 36*. **b)** An example of a PlaNet true negative and volunteer false positive. The dip observed is due to a background event. *TIC ID: 422506318, Sector: 42*. **c)** An example of a PlaNet false negative and volunteer true positive. *TIC ID: 460140752, Sector: 36*. **d)** An example of a PlaNet false positive and volunteer true negative. *TIC ID: 295317859, Sector: 39*.

B Implementation Details

B.1 Data

The composition of each of the training, validation, and test data splits is given in Table 2. Planets and EBs with TIC IDs that are multiples of 4 were chosen for synthetic data generation in the training set, and those with a remainder of 1 for the validation set. The rest were chosen for the test set.

When synthetic data was used, an equal proportion of synthetic transits to synthetic EBs was used in all cases. In the synthetic data composition experiments (Section 5), the model that used only synthetic data used 30% synthetic transits, 30% EBs, and 40% LCs which volunteers scored 0.0 which indicates they are unlike to contain a transit. On synthetic injection, we randomly selected a transit or EB from the relevant data split and point-wise multiplied it to the unnormalized base LC. We used base LCs that have a volunteer score of 0.0 such that they are unlikely to already contain a transit, but these may contain EBs. We further ensured that at least 80% of the section of the base LC where the transit is being injected into was not missing data. This is to prevent injecting a transit into a missing data region where it would not be visible and thus be mislabelled. LC noise was injected in a similar way using median-normalised injection curves which had a volunteer score of 0.0.

The signal to noise ratio (SNR) for synthetic data was calculated as the depth of the injected transit divided by the Combined Differential Photometric Precision (CDPP) of the base light curve. CDPP is the metric that defines the ease with which these weak terrestrial transit signatures can be detected. This is given at a series of durations (0.5, 1, 2 days). The closest one to the duration of the transits was

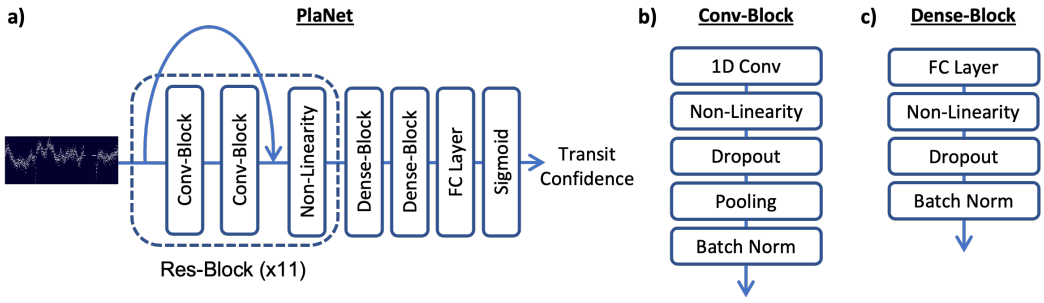

Figure 4: Simplified schematic of the proposed PlaNet architecture (inspired by Olmschenk et al. [17]). **a)** PlaNet features a series of convolutional blocks (Conv-Blocks) with residual connections [22], followed by two Dense-Blocks. **b)** A Conv-Block. **c)** A Dense-Block.

Table 2: Breakdown of the examples in each of the data splits used in this work. Strong volunteer scores are those that have an aggregated confidence over 0.5. The number of synthetic transits and eclipsing binaries used to generate the synthetic data is also given.

Data Split	Sectors	Total Light Curves	Non-zero Vol. Scores	Strong Vol. Scores	Synthetic Transits	Synthetic EBs
Training	10–29	367,417	58,721	8,311	278	65
Validation	30–35	118,344	29,850	2,974	316	69
Test	36–43	137,657	44,891	4,662	616	145

used to calculate the SNR. Unlike in PHT and other work [3, 18], we did not place a minimum SNR constraint on injected transits as we hope to be able to identify shallow transits. We did, however, place a maximum SNR constraint of 15, and a maximum duration of 4 days as in PHT.

B.2 Models

Figure 4 shows a schematic of the proposed PlaNet architecture. PlaNet consists of 11 residual blocks, each consisting of two convolutional blocks with a skip connection [22]. These are followed by 2 dense blocks and finally a fully-connected layer with sigmoid activation to output a transit confidence score between 0 and 1. We do not use a fully convolutional model as some degree of temporal awareness is required to identify EB false positives with multiple dips.

Conv-Blocks consists of a 1-D convolution, followed by a Leaky-ReLU non-linearity, dropout [25], a max-pooling operation and finally batch normalisation [26]. A generic Dense-Block consists of a linear layer, followed by a Leaky-ReLU, dropout and batch normalisation. Some operations in Conv and Dense-Blocks are not used depending on the layer of the network (Appendix B). In Res-Blocks, the second Conv-Block does not use a non-linearity. Instead the non-linearity is applied after the residual is added to its output. When pooling is used to downsample, a 1-D convolutional with kernel size of 1 and a stride equal to the pooling size is used to map the residual output to the same dimensions as the second Conv-Block output.

Models were implemented in *PyTorch* [27]. The main PlaNet model hyperparameters are given in Table 3. Architecture and optimization hyperparameters were chosen heuristically through minimising validation loss. Training runs on the full dataset took 15-24 hours on a single NVIDIA GeForce GTX 2080 Ti GPU.

Table 3: Hyperparameter configuration for PlaNet.

Hyperparameter	Value
Optimization	
Optimizer	AdamW [28]
L2 Weight Regularisation	0.01
Learning Rate	0.001
Batch Size	256
Early Stopping Patience (epochs)	300
Dropout Rate	0.1
Max Epochs	1000
Architecture	
Hidden Layer Non-linearities	Leaky-ReLU (slope=0.01)
Layers	[Res-Block (x11), Dense-Block (x2), Linear]
Kernel Size	[7, 5, 3, 3, 3, 3, 3, 3, 3, 3, 3, N/A, N/A, N/A]
Dropout (True/False)	[F, T, T, T, T, T, T, T, T, T, T, T, F, N/A]
Batch Normalisation (True/False)	[F, T, T, T, T, T, T, T, T, T, T, T, F, N/A]
Number of Out Channels/Units	[32, 32, 32, 64, 64, 128, 128, 128, 128, 128, 128, 1280, 256, 20]
Max Pooling Size	[1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, N/A, N/A, N/A]
Total Number of Parameters	1,054,889
Data	
Synthetic Transit Proportion	0.1
Synthetic EB Proportion	0.1
Training Augmentation Probability	0.1
Bin Factor (Max LC Length)	7 (2500)
Permute Fraction in Training	0.25
Delete Fraction in Training	0.1